IN THE CLAIMS:

1. (Currently Amended)A method for selecting a coredump disk from a set of disks owned by a failed server into which memory data ("coredump") of the failed server is stored, ~~the method~~ comprising ~~the steps of~~:

(a) identifying available disks that can receive a coredump;

(b) ordering the disks identified in step (a) so as to prefer disks that are least likely to be needed for normal service by the server;

(c) further ordering the disks so as to prefer disks that require a least amount of preparation to receive a coredump; and

(d) selecting as a best candidate for use as the coredump disk a first one of the disks in the order after steps (a), (b), (c) and (d).

2. (Original) The method as set forth in claim 1 further comprising, determining whether the selected coredump disk is formatted for use in normal service and if formatted, writing a "not-formatted" attribute to labels of the coredump disk.

3. (Original) The method as set forth in claim 1 further comprising, maintaining an attribute in a core region header of the coredump disk to indicate the current status of the coredump, the status including each of "complete coredump," "no coredump," "coredump in progress" and "coredump aborted."

4. (Original) The method as set forth in claim 3 further comprising, writing a "coredump in progress" signature to the coredump disk so as to prevent the coredump disk from being used for normal file service including updating the coredump status attribute to indicate the status of "coredump in-progress" and writing the updated "coredump in-progress" coredump status attribute to the core region header of the coredump disk.

5. (Original) The method as set forth in claim 1 further comprising providing pointers in the core region header that point to a file system region of the coredump disk in which the coredump is stored.

6. (Original) The method as set forth in claim 5 further comprising, writing the coredump to the file system region.

7. (Original) The method as set forth in claim 1 further comprising, where a spare disk is unavailable for selection as the coredump disk, providing coredump data to a core region in each of the set of disks in a distributed manner and, after one of either the rebooting by the failed filer or taking-over ownership of the set of disks by the taking-over ownership other filer, formatting and writing the data of the coredump to a root file system of either of the rebooted failed filer or the taking-over ownership other filer, respectively.

8. (Currently Amended) A method of identifying a coredump disk, during one of either rebooting by the failed filer or taking-over ownership the set of disks by another filer, from a set of disks owned by a failed filer into which memory data ("coredump") of the failed filer is stored, ~~the method~~ comprising ~~the steps of~~:

locating the coredump disk by reading labels on the disks of the set of disks to locate any of the disks that are non-formatted spare disks;

caching the labels for later use by a disk label assimilation process; and

reading a core region header in a respective label of the cached labels of each of the non-formatted spare disks to locate a coredump status attribute in the respective of the labels.

9. (Original) The method as set forth in claim 8 further comprising, freeing the cached labels after either one of the rebooting or taking-over ownership occurs and the disk label assimilation process completes.

10. (Original) The method as set forth in claim 8 further comprising, detecting a completed coredump in the coredump disk by locating a coredump status attribute in the coredump disk and thereafter formatting and writing data of the coredump to a root file system of either the rebooted failed filer or the taking-over ownership other filer.

11. (Original) The method as set forth in claim 10 further comprising, returning the coredump disk to a "hot" spare status from a status in which the coredump disk is unavailable for use in normal file service including updating the coredump status attribute to indicate a status of "no coredump" and writing the updated "no coredump" coredump status attribute data to the core region header of the coredump disk.

12. (Original) The method as set forth in claim 10 further comprising, writing a "coredump in progress" signature to the coredump disk so as to prevent the coredump disk from being used for normal file service including updating the coredump status attribute to indicate a status of "coredump in-progress" and writing the updated "coredump in-progress" coredump status attribute to the core region header of the coredump disk.

13. (Original) The method as set forth in claim 8 further comprising, where a spare disk is unavailable for selection as the coredump disk, providing coredump data to a core region in each of the set of disks in a distributed manner and, after one of either the rebooting by the failed filer or taking-over ownership of the set of disks by the taking-over ownership other filer, formatting and writing the data of the coredump to a root file system of either of the rebooted failed filer or the taking-over ownership other filer, respectively.

14. - 17.(Cancelled)

18. (Currently Amended) A method for selecting from a pool of candidate data storage devices owned by a server that includes a set of data storage devices with which normal file service by the server is performed, a best candidate data storage device for receiving

in a predetermined coredump storage space from the server memory contents (a "core-dump") comprising ~~the steps of~~:

identifying data storage devices in the pool of candidate data storage devices that are available and sized to receive the coredump; and

selecting the best candidate data storage device based upon a predetermined combination criteria including: (i) which data storage device of the pool of candidate data storage devices is adapted to complete reception of the coredump in the shortest time; (ii) which data storage device of the pool of candidate data storage devices is adapted to complete reception of the coredump with the least disruption of the normal file service; and (iii) which data storage device of the pool of candidate data storage devices is adapted to receive the coredump with minimal excess of storage space in the predetermined coredump storage region after reception is complete.

19. (Original)The method as set forth in claim 18 wherein the step of identifying further comprises selecting any available spare data storage device from the pool of candidate storage devices.

20. (Original) The method as set forth in claim 19 wherein the step of selecting further comprises:

(a) selecting from the identified spare data storage devices those having a core region, including a core region header;

(b) further selecting from the identified spare data storage devices in step (a) those having a file system region of sufficient space to receive the coredump; and

(c) ordering the identified spare data storage devices selected by steps (a) and (b) so as to prefer data storage devices least likely to be needed for normal file service by the file server;

(d) further ordering the identified spare data storage devices so as to prefer the disks requiring a least amount of preparation to receive the coredump; and

(e) selecting as the best candidate, the one of the identified data storage devices, which is the first data storage device yielded after the steps (a), (b), (c) and (d).

21. -23. (Cancelled)

24. (Currently Amended) A system for selecting from a pool of candidate data storage devices owned by a server that includes a set of data storage devices with which normal file service by the server is performed, a best candidate data storage device for receiving in a predetermined coredump storage space from the server memory contents (a "core-dump") comprising ~~the steps of~~:

an identifier that identifies data storage devices in the pool of candidate data storage devices that are available and sized to receive the coredump; and

a selector that selects the best candidate data storage device based upon a predetermined combination criteria including: (i) which data storage device of the pool of candidate data storage devices is adapted to complete reception of the coredump in the shortest time; (ii) which data storage device of the pool of candidate data storage devices is adapted to complete reception of the coredump with the least disruption of the normal file service; and (iii) which data storage device of the pool of candidate data storage devices is adapted to receive the coredump with minimal excess of storage space in the predetermined coredump storage region after reception is complete.

25. (Original) The system as set forth in 24 wherein the identifier is constructed and arranged to select any available spare data storage device from the pool of candidate storage devices.

26. (Original) The system as set forth in claim 25 wherein the selector is constructed and arranged to:

(a) select from the identified spare data storage devices those having a core region, including a core region header;

(b) further select from the identified spare data storage devices in step (a) those having a file system region of sufficient space to receive the coredump; and

(c) order the identified spare data storage devices selected by steps (a) and (b) so as to prefer data storage devices least likely to be needed for normal file service by the file server;

(d) further order the identified spare data storage devices so as to prefer the disks requiring a least amount of preparation to receive the coredump; and

(e) select as the best candidate, the one of the identified data storage devices, which is the first data storage device yielded after the selector performs (a), (b), (c) and (d).

27. – 29. (Cancelled)

30. (Previously Presented) A method, comprising:

identifying available disks of a group of disks that can receive a coredump; and

selecting a best candidate for use as a coredump disk, where the best candidate is least likely to be needed for normal service by a server, and requires the least amount of preparation to receive the coredump.

31. (Previously Presented) The method as set forth in claim 30, further comprising:

determining whether the selected coredump disk is formatted for use in normal service and if formatted, writing a "not-formatted" attribute to labels of the coredump disk.

32. (Previously Presented) The method as set forth in claim 30, further comprising:

maintaining an attribute in a core region header of the coredump disk to indicate the current status of the coredump, the status including each of "complete coredump," "no coredump," "coredump in progress" and "coredump aborted."

33. (Previously Presented) The method as set forth in claim 32, further comprising:

writing a "coredump in progress" signature to the coredump disk so as to prevent the coredump disk from being used for normal file service including updating the core-

dump status attribute to indicate the status of "coredump in-progress" and writing the updated "coredump in-progress" coredump status attribute to the core region header of the coredump disk.

34. (Previously Presented) The method as set forth in claim 32, further comprising:

providing pointers in the core region header that point to a file system region of the coredump disk in which the coredump is stored.

35. (Previously Presented) The method as set forth in claim 34, further comprising:

writing the coredump to the file system region.

36. (Previously Presented) An apparatus, comprising:

means for identifying available disks of a group of disks that can receive a core-dump; and

means for selecting a best candidate for use as a coredump disk, where the best candidate is least likely to be needed for normal service by a server, and requires the least amount of preparation to receive the coredump.

37. (Previously Presented) The apparatus as set forth in claim 36, further comprising:

means for determining whether the selected coredump disk is formatted for use in normal service and if formatted, writing a "not-formatted" attribute to labels of the core-dump disk.

38. (Previously Presented) The apparatus as set forth in claim 36, further comprising:

means for maintaining an attribute in a core region header of the coredump disk to indicate the current status of the coredump, the status including each of "complete core-dump," "no coredump," "coredump in progress" and "coredump aborted."

39. (Previously Presented) The apparatus as set forth in claim 38, further comprising:

means for writing a "coredump in progress" signature to the coredump disk so as to prevent the coredump disk from being used for normal file service including updating the coredump status attribute to indicate the status of "coredump in-progress" and writing the updated "coredump in-progress" coredump status attribute to the core region header of the coredump disk.

40. (Previously Presented) The apparatus as set forth in claim 38, further comprising:

means for providing pointers in the core region header that point to a file system region of the coredump disk in which the coredump is stored.

41. (Previously Presented) The apparatus as set forth in claim 40, further comprising:

means for writing the coredump to the file system region.